

WE CLAIM:

1. A method for representing header and footer structures in a markup language document, comprising:
  - determining properties corresponding to a mini-document that relates to at least one section of an application document;
  - mapping the properties of the mini-document into at least one of a markup language element, an attribute, and a value; and
  - storing the properties of the mini-document in the markup language document.
2. The method of Claim 1, further comprising determining whether the mini-document is one of a header and a footer.
3. The method of Claim 1, wherein mapping the properties further comprises mapping a type attribute that corresponds to the mini-document.
4. The method of Claim 3, wherein the type attribute corresponds to whether the mini-document occurs on a first page, odd pages, or even pages of the specified section of the application document.
5. The method of Claim 1, wherein determining the properties relating to the mini-document further comprises determining whether the mini-document corresponds to one of a context free chunk element, a paragraph element, and a table element.
6. The method of Claim 1, further comprising:
  - determining properties corresponding to an additional mini-document that relates to at least one section of the application document;
  - mapping the properties of the additional mini-document into at least one of a markup language element, an attribute, and a value; and

storing the properties of the additional mini-document in the markup language document.

7. The method of Claim 1, further comprising:  
determining whether properties associated with all mini-documents of the application document have been stored in the markup language document; and  
processing further mini-documents when the properties associated with all mini-documents have not been stored in the markup language document.

8. The method of Claim 1, wherein the properties of the mini-document stored in the markup language document are understood by an application that understands the markup language when the mini-document is not native to the application.

9. The method of Claim 1, wherein the markup language document is manipulated on a server to substantially reproduce the mini-document of the application document notwithstanding the presence of an application that generated the markup language document.

10. A computer-readable medium for representing headers and footers in a markup language document, comprising:  
determining properties relating to a mini-document used within a word-processing document;  
determining whether the mini-document is one of a header and a footer;  
writing the properties into at least one of a markup language element, an attribute, and a value; and  
storing the properties in the markup language document such that the headers and footers of the word-processing document are substantially maintained when the markup language document is parsed by an application.

11. The computer-readable medium of Claim 10, wherein the markup language document is manipulated on a server to substantially reproduce the mini-document of the word-processing document notwithstanding the presence of an application that generated the markup language document.

12. The computer-readable medium of Claim 10, wherein the properties of the mini-document stored in the markup language document are understood by an application that understands the markup language when the mini-document is not native to the application.

13. The computer-readable medium of Claim 10, wherein mapping the properties further comprises mapping a type attribute that corresponds to the mini-document.

14. The computer-readable medium of Claim 13, wherein the type attribute corresponds to whether the mini-document occurs on a first page, odd pages, or even pages of the specified section of the word-processing document.

15. The computer-readable medium of Claim 10, wherein determining the properties relating to the mini-document further comprises determining whether the mini-document corresponds to one of a context free chunk element, a paragraph element, and a table element.

16. The computer-readable medium of Claim 10, further comprising:  
determining properties corresponding to an additional mini-document that relates to at least one section of the word-processing document;  
mapping the properties of the additional mini-document into at least one of a markup language element, an attribute, and a value; and  
storing the properties of the additional mini-document in the markup language document.

17. The computer-readable medium of Claim 10, further comprising:  
determining whether properties associated with all mini-documents of the word-processing document have been stored in the markup language document; and  
processing further mini-documents when the properties associated with all mini-documents have not been stored in the markup language document.

18. A system for representing header and footer information in a markup language document, comprising:  
an application that is configured to:  
determine properties relating to a mini-document included in at least one section of an application document;  
determine whether the mini-document is one of a header and a footer;  
map the properties into at least one of a markup language element, an attribute, and a value; and  
store the properties in the markup language document; and  
a validation engine configured to validate the markup language document.

19. The system of Claim 18, wherein the application is further configured to:  
determine properties corresponding to an additional mini-document that relates to at least one section of the application document;  
map the properties of the additional mini-document into at least one of a markup language element, an attribute, and a value; and  
store the properties of the additional mini-document in the markup language document.

20. The system of Claim 18, wherein the application is further configured to:  
determine whether properties associated with all mini-documents of the application document have been stored in the markup language document; and

process further mini-documents when the properties associated with all mini-documents have not been stored in the markup language document.

21. The system of Claim 18, wherein the properties of the mini-document stored in the markup language document are understood by an additional application that understands the markup language when the mini-document is not native to the additional application.

22. The system of Claim 18, wherein the markup language document is manipulated on a server to substantially reproduce the mini-document of the application document notwithstanding the presence of the application that generated the markup language document.